



TRABAJO FIN DE GRADO
GRADO EN INGENIERIA INFORMATICA

Título

Subtítulo

Francisco de Asís Carrasco Conde

Autor

Francisco de Asís Carrasco Conde

Director

María José Rodríguez Fórtiz



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

—
Granada, Junio de 2025

Título
Subtítulo

Francisco de Asís Carrasco Conde

Palabras clave: *software libre*

Resumen

Same, but in English

Student's name

Keywords: *open source, floss*

Abstract

D. **María José Rodríguez Fórtiz**, Profesor(a) del Departamento de Lenguajes y Sistemas Informáticos

Informo:

Que el presente trabajo, titulado ***Nombre de la App***, ha sido realizado bajo mi supervisión por **Francisco de Asís Carrasco Conde**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a Junio de 2025.

El/la director(a)/es:

María José Rodríguez Fórtiz

Agradecimientos

Índice general

1. Introducción	17
1.1. Contexto	17
1.2. Justificación y Motivación	18
1.3. Objetivos e Hipótesis	19
1.4. Estructura de la Memoria	20
2. Estado del arte	21
2.0.1. Descripción de dominio del problema	21
2.0.2. Metodologías potenciales a aplicar	21
2.0.3. Tecnologías potenciales para usar	21
2.0.4. Trabajos relacionados	21
3. Descripción del problema	23
4. Planificación	25
4.1. Metodología utilizada	25
4.2. Temporización	25
4.3. Seguimiento del desarrollo	26
5. Análisis del problema	37
6. Implementación	39

7. Conclusiones y trabajos futuros	41
-------------------------------------------	-----------

Índice de figuras

1.1. Descripción de la imagen	19
4.1. Planificación Sprint 1	28
4.2. Subtareas Sprint 1	28
4.3. Fin Sprint 1	29
4.4. Subtareas completadas Sprint 1	29
4.5. Pantalla nueva	30
4.6. Pantalla antigua	30
4.7. Comparación entre la pantalla nueva y la anterior	30
4.8. Planificación Sprint 2	31
4.9. Subtareas Sprint 2	31
4.10. Fin Sprint 2 y subtareas	32
4.11. Planificación Sprint 3	33

Índice de tablas

Capítulo 1

Introducción

1.1. Contexto

Esta aplicación va a estar orientada a hacer de soporte a deportistas, tanto experimentados como no. Un ejercicio, es la repetición de un movimiento varias veces para estimular uno o varios músculos. Los entrenamientos, entendámoslos como la colección de distintos ejercicios destinados a entrenar una parte o varias del cuerpo. Una serie, es cuando dentro de un ejercicio, repetimos ese movimiento un número determinado de veces y se realiza un descanso, sin cambiar de ejercicio. Una repetición, es la realización del movimiento de ese ejercicio en una serie, es decir, si estoy haciendo flexiones, hago 12, descanso, hago 10, descanso, hago 9 y ya no hago más flexiones, dentro de mi entrenamiento se vería así:

Rutina de entrenamiento 1:

- Ejercicio 1
 - Serie 1: X repeticiones
 - Serie 2: X repeticiones
- Flexiones
 - Serie 1: 12 repeticiones
 - Serie 2: 10 repeticiones
 - Serie 3: 9 repeticiones
- Ejercicio 3

- Serie 1: X repeticiones
- Serie 2: X repeticiones

1.2. Justificación y Motivación

Algunas de las aplicaciones que existen, para mediciones de constantes relacionadas durante el entrenamiento físico, suelen ser de pago. Ejemplos:

- MyFitnessPal: €9.99
- Whoop: €28
- Fitbit Premium: €9.99
- Apple Fitness+: €9.99
- Strava Premium: €5.99

Es verdad que algunas tienen versión gratuita, pero no ofrecen la totalidad de sus funcionalidades, de hecho, estas versiones suelen ser extremadamente limitadas. Otras aplicaciones como Garmin Connect, si que son gratuitas, pero te obligan a ceñirte a un smartwatch de la marca Garmin, los cuales su precio no baja de los 300€.

La app que **estoy** presentando daría la mayoría de funcionalidades de pago de una manera **más barata que el resto** y aporta su funcionalidad de medir de forma personalizada el rendimiento del deportista. **También se añade la IA** que permite aconsejar al usuario.

Aquí una tabla resumiendo funcionalidades:

Funciones	Caliber(Lite)	Caliber(Premium)	Mi app
Visualizar la fuerza máxima	No	Si	Si
Visualizar el balance de fuerza	No	Si	Si
Librería de ejercicios que proporciona la app	Si	Si	No
Registra que rutina de ejercicios se hace por día	Si	Si	Si
Nutrición customizada para cada usuario según sus objetivos	No	Si	No
Indicaciones de como incrementar peso en los ejercicios	No	Si	No
Establecer metas ejercicios	No	Si	Si
Crear tus propios ejercicios	Si	Si	Si
Crear tus propias rutinas	Si	Si	Si
Subir fotos tuyas para medir tu progreso	No	Si	No
Sugiere ejercicios a sustituir por otros en caso del usuario necesitarlo	No	Si	El usuario a de especificarlo 1*
Registra el peso y estatura del usuario por día	Si	Si	Si
Muestra una gráfica de progreso	Si	Si	Si
Accesible	No(Demasiados scrollables y mala paleta de colores)		Si
Uso offline	Si(Hay que activarlo)		Si

Figura 1.1: Descripción de la imagen

1.3. Objetivos e Hipótesis

Los objetivos que planteo alcanzar son:

- Corrección de ejercicios durante su realización
- Evaluación del entrenamiento del usuario en base a sus objetivos
- Ofrecer consejo al usuario mediante una IA
- Recopilar una buena cantidad de datos sobre los entrenamientos del usuario
- Permitir a los usuarios de la app compartir sus entrenamientos entre ellos

- Customizar tus entrenamientos y ejercicios

1.4. Estructura de la Memoria

Explica cómo está estructurada la memoria del trabajo, mencionando brevemente los capítulos y lo que cada uno aborda.

Capítulo 2

Estado del arte

2.0.1. Descripción de dominio del problema

2.0.2. Metodologías potenciales a aplicar

2.0.3. Tecnologías potenciales para usar

En el front-end, mis favoritas según lo que he estado investigando, serían los framework Flutter y React Native. Al final me he decidido por Flutter, dado que tengo más experiencia con el.

En el back-end usaré Java, ya que es un lenguaje que domino ampliamente.

2.0.4. Trabajos relacionados

El software libre y sus licencias [1] ha permitido llevar a cabo una expansión del aprendizaje de la informática sin precedentes.

Capítulo 3

Descripción del problema

Capítulo 4

Planificación

4.1. Metodología utilizada

Para el desarrollo de esta app, usaré una metodología *ágil* tipo **SCRUM**. He decidido usar esta porque me permite corregir fallos en la velocidad de diseño y/o planificación de forma eficiente y sin dañar el producto final.

Las reuniones con la tutora *serán las **sprint reviews***.

4.2. Temporización

La temporización se realizó el día 25 de marzo de 2025. La entrega del producto (este TFG) está prevista para el 16 de junio de 2025, es decir, 83 días, o lo que es lo mismo, casi 12 semanas. *Si un sprint dura 2 semanas*, habrá 6 sprints hasta la entrega final.

La iteración 0 se dedicará *al diseño de pantallas y al repaso de las funcionalidades de la app*. La idea inicial es *dividir la app en varios módulos* y centrar cada sprint en uno de ellos:

- *Diagramas* de la app y ejercicios
- Rutinas, usuarios y sesión
- Datos que ingresa el usuario
- Flujo de entrenamiento

- Inteligencia Artificial (IA)
- Smartwatch

4.3. Seguimiento del desarrollo

Iteración 0

En esta primera iteración me centré en completar los diseños de la app, priorizando su accesibilidad. También concreté el *product backlog*, compuesto por 25 historias de usuario. Algunas incluyen tareas secundarias:

- SCRUM-1** Registrar peso por día
- SCRUM-2** Establecer peso objetivo
- SCRUM-3** Insertar/Borrar/Modificar ejercicio de la lista
- SCRUM-4** Buscar rutina en la lista del usuario
- SCRUM-5** Sustituir un ejercicio por otro en la rutina
- SCRUM-6** Insertar/Borrar/Modificar rutina
- SCRUM-7** Poner una meta en cada ejercicio
- SCRUM-8** Graficar los datos de los ejercicios
- SCRUM-9** Revisar datos para ver si el descanso es necesario
- SCRUM-10** Enseñar datos de una rutina a descargar
- SCRUM-11** Compartir mi rutina
- SCRUM-12** Guardar las repeticiones y series de todos los ejercicios de un entrenamiento
- SCRUM-13** Valorar el entrenamiento en base a la marca actual y la meta del usuario
- SCRUM-14** Monitorizar pulso en tiempo real
- SCRUM-15** Medir pulso en reposo y compararlo con datos de ejercicios
- SCRUM-16** Avisar de anomalías en el pulso de forma suave
- SCRUM-17** Obtener calorías quemadas

SCRUM-18 Comprobar el equilibrio nervioso del usuario

SCRUM-19 Ejecutar el flujo del entrenamiento

SCRUM-20 Conectar con la IA para iniciar diálogo

SCRUM-21 Crear/Borrar usuario

SCRUM-22 Resumir datos

SCRUM-23 Iniciar/Cerrar sesión

SCRUM-24 Medir SpO2

SCRUM-25 Interpretar constantes

Sprint Review 0

Durante esta revisión de sprint se realizaron mejoras de diseño, como la incorporación de iconos en todos los botones para mejorar la accesibilidad. Se revisaron los títulos de las ventanas, cambiando aquellos que no eran lo suficientemente claros.

Además, se añadieron nuevas historias al *product backlog*:

- Copiar rutina
- Añadir meta por parámetro
- Solicitar permiso al usuario antes de enviar datos a la IA
- Enviar datos del entrenamiento actual y anteriores a la IA
- Conectar/Desconectar con la IA

Iteración 1

Al inicio de esta iteración detecté la ausencia de historias para el diseño de diagramas de la app y la base de datos. Por ello, se añadió la siguiente historia:

SCRUM-26 Diseñar las tablas de la base de datos

Durante esta iteración me familiaricé con Flutter y sus herramientas, como la base de datos local *SQLite*.

☐ Tablero Sprint 1 25 mar – 8 abr (7 incidencias) 8 0 0 Completar sprint ...

<input checked="" type="checkbox"/>	SCRUM-39	Buscar ejercicio en lista	TAREAS POR ...	1	
<input checked="" type="checkbox"/>	SCRUM-7	Insertar/Borrar/Modificar ejercicio de la lista	TAREAS POR ...	2	
<input checked="" type="checkbox"/>	SCRUM-34	Diseñar tablas para la BD de los ejercicios y marcas	TAREAS POR ...	1	
<input checked="" type="checkbox"/>	SCRUM-61	Hacer la IU de la lista de ejercicios	TAREAS POR ...	1	
<input checked="" type="checkbox"/>	SCRUM-80	IU de creación de ejercicio	TAREAS POR ...	1	
<input checked="" type="checkbox"/>	SCRUM-81	Pop up de confirmación	TAREAS POR ...	1	
<input checked="" type="checkbox"/>	SCRUM-83	IU datos ejercicio	TAREAS POR ...	1	

+ Crear incidencia

Figura 4.1: Planificación Sprint 1

Insertar/Borrar/Modificar ejercicio de la lista

+ Añadir @ Aplicaciones

Descripción

Editar descripción

Incidencias secundarias

0 % hecho

T...	Clave	Resumen	P...	P...	Estado	
	SCRUM-35	Guardar en la BD	=			TAREAS POR ...
	SCRUM-36	Borrar de la BD	=			TAREAS POR ...
	SCRUM-37	Hacer notorios los cambios en vivo	=			TAREAS POR ...
	SCRUM-38	Modificar en la BD	=			TAREAS POR ...

Figura 4.2: Subtareas Sprint 1

SQLite

Es una base de datos ligera, autocontenida y de código abierto. Usa el mismo lenguaje de consultas que SQL, lo que la hace ideal para la app. Además, permite transportar toda la BD en un *.db*, lo cual puede facilitar funcionalidades como copias de seguridad descargables desde la nube.

Sprint Review 1

Las primeras iteraciones tienden a ser más lentas debido a la fase inicial del desarrollo. No se completó la totalidad del sprint; quedó pendiente la subtask de modificar ejercicio en la base de datos. Aun así, las expectativas son positivas,

ya que se espera un aumento en la velocidad de desarrollo. No obstante, en lo que se lleva desarrollado han aparecido pocas problemáticas

Task ID	Task Name	Status	Priority	Assignee
SCRUM-39	Buscar ejercicio en lista	FINALIZADA	1	[User]
SCRUM-7	Insertar/Borrar/Modificar ejercicio de la...	TAREAS POR H...	2	[User]
SCRUM-34	Diseñar tablas para la BD de los ejercicios ...	FINALIZADA	1	[User]
SCRUM-61	Hacer la IU de la lista de ejercicios	FINALIZADA	1	[User]
SCRUM-80	IU de creación de ejercicio	FINALIZADA	1	[User]
SCRUM-81	Pop up de confirmación	FINALIZADA	1	[User]
SCRUM-83	IU datos ejercicio	FINALIZADA	1	[User]

Figura 4.3: Fin Sprint 1

Insertar/Borrar/Modificar ejercicio de la lista

+ Añadir @ Aplicaciones

Descripción
Editar descripción

Incidencias secundarias 75 % hecho

T...	Clave	Resumen	P...	P...	Estado
[Icon]	SCRUM-35	Guardar en la BD	[Icon]	[User]	FINALIZADA
[Icon]	SCRUM-36	Borrar de la BD	[Icon]	[User]	FINALIZADA
[Icon]	SCRUM-37	Hacer notorios los cambios en vivo	[Icon]	[User]	FINALIZADA
[Icon]	SCRUM-38	Modificar en la BD	[Icon]	[User]	TAREAS POR ...

Figura 4.4: Subtareas completadas Sprint 1

Iteración 2

Esta iteración se centró en el desarrollo del sistema de sesiones, la API de la aplicación, el backend básico del servidor y la sección de rutinas.



Figura 4.5: Pantalla nueva

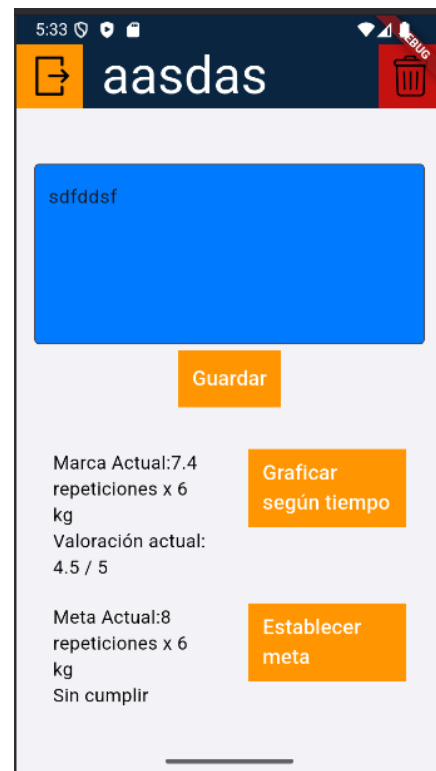


Figura 4.6: Pantalla antigua

Figura 4.7: Comparación entre la pantalla nueva y la anterior

Sistema de sesiones

Se implementó un sistema de autenticación basado en tokens. Cuando el usuario verifica su identidad, recibe un token que se guarda en el dispositivo y se usa para validar el acceso posterior. Esto evita que usuarios no autorizados suban contenido haciéndose pasar por otros.

El backend está desarrollado en *Node.js* y utiliza una base de datos SQL para gestionar las contraseñas de los usuarios. Este mismo backend se usará para las funcionalidades de las siguientes iteraciones.

API de la app

La comunicación entre la app y el servidor se realiza mediante peticiones HTTP. En el futuro, estas podrán migrarse a HTTPS para proteger operaciones

Task ID	Task Name	Progress	Assignee
SCRUM-22	Crear/Borrar mi usuario	TAREAS POR ...	1
SCRUM-25	Iniciar/Cerrar sesion	TAREAS POR ...	1
SCRUM-6	Insertar/Borrar/Modificar rutina	TAREAS POR ...	2
SCRUM-4	Buscar rutina en la lista del usuario	TAREAS POR ...	1
SCRUM-63	Crear IU de la lista de las rutinas	TAREAS POR ...	1
SCRUM-65	Crear IU de pantalla inicial	TAREAS POR ...	2
SCRUM-66	Crear IU menu principal	TAREAS POR ...	1
SCRUM-84	IU pop up rutinas	TAREAS POR ...	1
SCRUM-85	IU lista ejercicios modificable o no	TAREAS POR ...	1
SCRUM-86	IU pop up crear rutina	TAREAS POR ...	1

Figura 4.8: Planificación Sprint 2

Crear IU de pantalla inicial

+ Añadir

👤 Aplicaciones

Descripción

Editar descripción

Actividades secundarias

🔍 ... +

0 % hecho

T...	Clave	Resumen	P...	P...	Estado	
🔖	SCRUM-67	Crear usuario	==	👤	TAREAS POR ...	
🔖	SCRUM-68	Iniciar sesion	==	👤	TAREAS POR ...	
🔖	SCRUM-69	Crear usuario 2	==	👤	TAREAS POR ...	

Figura 4.9: Subtareas Sprint 2

sensibles. Por ahora se mantiene HTTP para facilitar el depurado.

Cambios importantes

Se rediseñó la implementación de las pantallas con listas. En lugar de una única clase con condiciones para la visibilidad de elementos, se optó por crear clases específicas para cada tipo de lista.

Ventaja: El código es más limpio, escalable y modular. Si hay que modificar un tipo de lista, los demás no se ven afectados.

Sprint Review 2

Se sugirieron mejoras menores como el ajuste de tamaños e iconos en los botones de la sección de creación de rutinas.

También se modificó la funcionalidad de compartir rutinas. Ahora todas son modificables, eliminando la distinción entre rutinas descargadas (antes no modificables) y creadas (modificables). Esto mejora la experiencia del usuario: si desea volver a una rutina original, simplemente la puede volver a buscar y descargar.

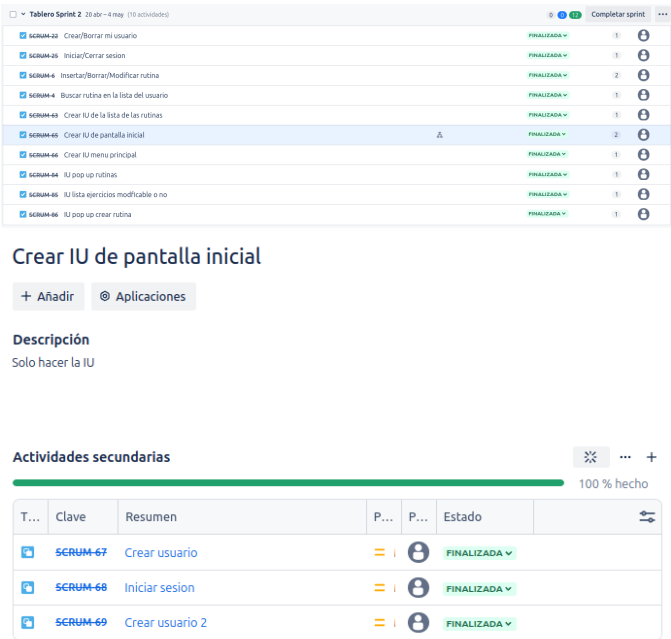


Figura 4.10: Fin Sprint 2 y subtareas

Iteración 3

Esta iteración se centró en desarrollar la funcionalidad para compartir rutinas entre usuarios, incluyendo la subida, visualización y descarga de rutinas. Además, se comenzó a implementar la funcionalidad de resumen de datos para optimizar el uso de la memoria local.

Task ID	Task Name	Assignee	Label
SCRUM-2	Establecer peso objetivo	[User Icon]	TAREAS POR H... 1
SCRUM-62	Crear IU del perfil del usuario	[User Icon]	TAREAS POR H... 1
SCRUM-26	Copiar rutina	[User Icon]	TAREAS POR H... 1
SCRUM-11	Enseñar datos de una rutina a descargar	[User Icon]	TAREAS POR H... 1
SCRUM-12	Compartir mi rutina	[User Icon]	TAREAS POR H... 1
SCRUM-64	Crear IU de buscar rutinas para descargar	[User Icon]	TAREAS POR ... 2
SCRUM-10	Revisar datos para ver si el descanso es necesario	[User Icon]	TAREAS POR ... 2
SCRUM-23	Resumir datos	[User Icon]	TAREAS POR ... 2
SCRUM-77	IU del pop up de resumir datos	[User Icon]	TAREAS POR ... 1
SCRUM-87	IU desplegable rutina	[User Icon]	TAREAS POR ... 1

Figura 4.11: Planificación Sprint 3

Peso objetivo

El peso objetivo es una meta que define el usuario y se almacena en memoria. Esta meta se representará en la gráfica de evolución del peso del usuario, y se generará una alerta cuando dicha meta sea alcanzada. La visualización aún no ha sido implementada completamente.

Cambios durante el desarrollo

Durante esta iteración surgieron cambios en la forma de visualizar y gestionar las rutinas compartidas y almacenadas:

- Ahora, al descargar una rutina, se añade el nombre del creador al nombre de la rutina, facilitando la distinción entre rutinas propias y descargadas.
- Cuando hay conflicto de nombres al crear una rutina, se genera un nombre alternativo del tipo Nombre(n), siendo n un número incremental.
- Se ha implementado un selector entre “Mis rutinas” y “Compartidas” para facilitar la visualización de las rutinas subidas a la nube por el usuario.
- Se reemplazó el filtro integrado en la ventana de búsqueda por un selector emergente que permite buscar rutinas por nombre o por nombre de usuario.
- La interfaz de descarga y visualización de rutinas ahora es un cuadro emergente (pop-up) en lugar de un desplegable.

Imprevistos y problemas en el desarrollo

Se identificaron dos errores principales de planificación:

Error 1: Se planificó implementar la funcionalidad de resumir datos sin haber completado la obtención de datos.

Solución: Replanificar los sprints.

Error 2: Se subestimó la complejidad del sistema de rutinas compartidas, especialmente en la resolución de conflictos de nombres.

Solución: Aplicar nombres combinados (nombre + creador + copia) en **memoria local** e identificadores *autoincrementales* en la nube.

Valor añadido a la app

Los imprevistos y problemas surgidos durante el desarrollo han permitido detectar *bugs*, errores de diseño y carencias funcionales que de otro modo podrían haber pasado desapercibidos. Gracias a ello, se han podido proponer nuevas funcionalidades y mejorar las ya existentes, lo cual contribuye significativamente a aumentar la calidad global de la aplicación.

Algunas de las funcionalidades propuestas como mejora son:

- Verificar el token del usuario antes de permitir la subida de rutinas, para evitar plagios.
- Al seleccionar la opción de borrar usuario del dispositivo, **eliminar también la base de datos local asociada.**
- Crear una base de datos local independiente para cada nuevo usuario creado en un dispositivo.
- Posibilidad de editar el nombre de una rutina.
- Marcar los ejercicios eliminados con una *flag*, para evitar que se inicien rutinas que los contengan.
- Marcar a los usuarios permanentemente eliminados con una *flag*, para proceder a eliminarlos en los dispositivos.

Estas funcionalidades están pensadas para aportar mayor **seguridad**, **integridad** y **calidad** al producto final.

4.3.0.1. Base de datos del backend

La base de datos utilizada en el backend está basada en *MySQL*. En ella se almacenan los usuarios junto con sus contraseñas, así como los ejercicios y las rutinas. Las rutinas están vinculadas al usuario que las creó, y los ejercicios se asocian a las rutinas a las que pertenecen. Si un usuario decide eliminar su cuenta de forma permanente, las rutinas y ejercicios relacionados se eliminarán en cascada.

Al descargar una rutina, los ejercicios asociados se guardan en la memoria local del usuario como si fueran de su propiedad. Es decir, el usuario puede modificarlos libremente sin restricciones.

Capítulo 5

Análisis del problema

Capítulo 6

Implementación

La implementación del software se ha dividido en hitos. Estos, han sido definidos en Github y cada uno de ellos contiene un grupo de *issues* que se corresponden con las distintas mejoras que se han ido incorporando al software a lo largo de su desarrollo.

Capítulo 7

Conclusiones y trabajos futuros

Bibliografía

- [1] Free Software Foundation. GNU General Public License. <http://www.gnu.org/licenses/gpl.html>.